

Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Mobilný sprievodca ZOO
Mobile ZOO Guide

2014

Lukáš Řezanina

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Lukáš Řezanina**
Studijní program: B2647 Informační a komunikační technologie
Studijní obor: 2612R025 Informatika a výpočetní technika
Téma: **Mobilní průvodce Zoo
Mobile Zoo Guide**

Zásady pro vypracování:

Cílem práce je vytvořit multiplatformní mobilní aplikaci, jež bude sloužit jako průvodce zoologickou zahradou.

Aplikace bude obsahovat:

1. Mapu areálu.
2. Přehled expozic a zvířat s konkrétními popisy a fotografiemi.
3. Aktuální informace o dění v ZOO.
4. Další dle pokynů vedoucího práce.

V aplikaci bude využito geolokace pro notifikaci uživatele na zajímavosti v jeho okolí. Obsah aplikace bude možné aktualizovat z webového serveru. Součástí práce bude představení a srovnání možností frameworků pro tvorbu multiplatformních mobilních aplikací.

Seznam doporučené odborné literatury:

- [1] WARGO, John M. PhoneGap essentials: building cross-platform mobile apps. Upper Saddle River, NJ: Addison-Wesley, c2012, xxiv, 359 p. ISBN 03-218-1429-0.
- [2] SHOTTS, Kerri. Phonegap hotshot. 1. publ. S.l.: Packt Publishing Limited, 2013. ISBN 978-184-9519-403.
- [3] PILGRIM, Mark. HTML5: up and running. 1st ed. Sebastopol, CA: O'Reilly, 2010. ISBN 05-968-0602-7.
- [4] NEIL, Theresa. Mobile design pattern gallery. 1st ed. Sebastopol, CA: O'Reilly, c2012, xii, 261 p. ISBN 14-493-1432-5.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Janoušek**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2014



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prehlásenie študenta

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal

V Ostrave dňa: *18. júla 2014*


.....
podpis študenta

Pod'akovanie

Týmto by som rád pod'akoval vedúcemu mojej bakalárskej práce, Ing. Janovi Janouškovi, za odbornú, metodickú pomoc a taktiež za trpezlivosť, ktorú mal po celú dobu spracovávaní tejto bakalárskej práce. Taktiež by som rád pod'akoval svojim rodičom za nie len psychickú podporu.

Abstrakt

Témou tejto bakalárskej práce je multiplikované programovanie aplikácií. V úvodných kapitolách práca popisuje širší kontext tvorby mobilných aplikácií a tiež obsahuje zhodnotenie súčasnej situácie ohľadne ich tvorby. V nasledujúcej časti popisuje možnosť použitia multiplatformového programovania mobilných aplikácií. Taktiež zahrňuje posúdenie výhod a nevýhod pri multiplatformovom programovaní. Následne práca popisuje praktickú implementáciu multiplatformového programovania s využitím HTML5, CSS3 a JavaScript. V ďalších kapitolách je uvedený prakticky postup pri tvorbe multiplatformovej aplikácie s názvom "Mobilný sprievodca ZOO". Aplikácia bola vytvorená pomocou softwarového rámca PhoneGap, ktorý je v práci popísaný. Záver mojej obsahuje testovanie aplikácie na rôznych mobilných platformách.

Kľúčové slová

multiplatformové programovanie, mobilné aplikácie, PhoneGap, mobilné platformy, Android

Abstract

A topic of this thesis is multiplatform programming of applications. In the first chapters thesis describes wide context of creating mobile applications and also we can find the assessment of the current situation in mobile applications programming. The following part describes the possibilities in their implication. Work also includes advantages and disadvantages. Subsequently, the work describes the practical implementation with HTML5, CSS and JavaScript. In the next chapters of Mobile ZOO Guide I describe every step in programming this kind of multiplatform application. Mobile ZOO Guide was created by PhoneGap framework which is previously described in the thesis. At the end of the thesis are described all kinds of testing we have made on different platforms.

Key words

multiplatform programming, mobile applications, PhoneGap, mobile platforms, Android

Zoznam použitých skratiek

Skratka	Význam
HTML	Hypertextový značkovací jazyk
HTML5	Nová verzia hypertextového značkovacieho jazyka
CSS	Kaskádové štýly
SDK	Súbor nástrojov pre vývoj softvéru
GPS	Globálny lokalizačný systém
API	Rozhranie pre programovacie jazyky
ADT	Android nástroje pre vývoj
JDT	Java nástroje pre vývoj
OHA	Aliancia technologických a mobilných firiem
XHR	Webový skriptovací jazyk
DOM	Objektový model dokumentu k XML
SHA1	Rozšírená hašovacia funkcia
GUI	Grafické rozhranie
.NET	Súbor technológií v softvérových produktoch

Obsah

Úvod.....	- 10 -
1 Mobilné platformy a vývoj aplikácií	- 11 -
1.1 Android	- 11 -
1.1.1 História	- 11 -
1.1.2 Vývoj aplikácií	- 11 -
1.2 iOS.....	- 12 -
1.2.1 História	- 12 -
1.2.2 Vývoj aplikácií	- 12 -
1.3 Microsoft Windows Phone	- 13 -
1.3.1 História	- 13 -
1.3.2 Vývoj aplikácií	- 13 -
1.4 Ostatné mobilné platformy	- 14 -
1.4.1 Symbian.....	- 14 -
1.4.2 Samsung Bada	- 14 -
1.4.3 Blackberry	- 15 -
2 Multiplatformové aplikácie verzus natívne aplikácie	- 16 -
2.1 Výber architektúry.....	- 16 -
2.1.1 Natívne aplikácie.....	- 16 -
2.1.2 Webové aplikácie	- 16 -
2.1.3 Multiplatformové - hybridné aplikácie.....	- 17 -
3 Multiplatformové softvérové rámce.....	- 19 -
3.1 PhoneGap	- 19 -
3.2 Marmalade.....	- 19 -
3.3 Appcelerator Titanium	- 20 -
3.4 Ostatné softvérové rámce	- 20 -
3.5 Porovnanie a hodnotenie	- 21 -
4 Návrh multiplatformovej aplikácie v softvérovom rámci PhoneGap.....	- 22 -
4.1 Predstavenie aplikácie	- 22 -
4.2 Použité technológie a softvérové rámce	- 22 -

4.3	Použité zásuvné moduly	- 22 -
4.3.1	Google Android Maps API V2	- 22 -
4.3.2	Google Maps API V3	- 23 -
4.3.3	Push notifikácia	- 24 -
4.3.4	jQuery Mobile	- 25 -
4.4	Ciele aplikácie	- 25 -
5	Implementácia aplikácie Mobilný sprievodca v ZOO	- 26 -
5.1	Prvotný návrh	- 26 -
5.2	Vývoj aplikácie	- 26 -
5.3	Odporúčaný vývoj aplikácie	- 27 -
5.3.1	Analýza	- 27 -
5.3.2	Vývoj	- 27 -
5.3.3	Pripojenie	- 30 -
5.3.4	Prehľad konečnej aplikácie	- 30 -
5.4	Kompilácia	- 31 -
6	Testovanie	- 33 -
6.1	Testovanie pri vývoji	- 33 -
6.2	Užívateľské testovanie	- 34 -
6.3	Zhodnotenie vlastností aplikácie na rôznych platformách	- 34 -
	Záver	- 36 -
	Použitá literatúra	- 37 -
	Zoznam príloh	XXXVIII

Úvod

Bakalárska práca na tému Mobilný sprievodca ZOO bola mojou prvou voľbou. Aj keď moje vedomosti v tejto sfére multiplatformových aplikácií boli z počiatku minimálne, enormný potenciál, aký tento sektor informatiky skrýva, ma presvedčil. Tvorba mobilných aplikácií je na trhu veľmi žiadanou službou a podobne ako pred desiatimi rokmi všetky organizácie, spoločnosti či malé firmy museli vstúpiť do sveta informačných technológií so svojou webovou prezentáciou, teraz je na rade prezentácia mobilnou aplikáciou.

Multiplatformové, nazývané aj hybridné aplikácie, zažívajú obrovský úspech v posledných rokoch. Myšlienka nadviazať na dlhoročnú prácu vývojárov webových stránok, nadviazať na ich vedomosti a sprístupniť im hardwarové možnosti mobilných zariadení prináša pozitívne výsledky už v prvých rokoch vývoja. Softvérové rámce pre tvorbu hybridných aplikácií vznikajú bleskovo a veľké spoločnosti investujú nemalé peniaze do ich vývoja.

Cieľom mojej bakalárskej práce je zorientovať sa vo svete hybridných mobilných aplikácií a pokúsiť sa využiť ich potenciál. Výsledkom práce bude mobilná aplikácia Mobilný ZOO sprievodca, ktorá má pomáhať. Pomáhať najmladším, mladým, alebo aj tým starším užiť si ZOO v Ostrave a všetky jej vymoženosti. Vedieť, kam ísť, akú akciu nezmeškať, kde si oddýchnuť, či absorbovať nové zaujímavé informácie, ktoré obohatia zážitok zo ZOO.

1 Mobilné platformy a vývoj aplikácií

1.1 Android

Android je rozsiahla voľne dostupná platforma určená najmä pre mobilné zariadenia. Postavená je na základe Linux, distribuovaná pod licenciou Apache IMIT. Zdrojové kódy boli dané k dispozícii od nízkoúrovňových linuxových modulov, cez natívne knižnice, po vysokoúrovňové aplikačné rámce, čo umožňuje výrobcovi mobilných zariadení systém do značnej miery upravovať.

1.1.1 História

Myšlienka vytvoriť nový operačný systém Android sa v hlavách zakladateľov Andyho Rubina, Richa Minera, Nicka Searsa a Chrisa Whitea zrodila už pred viac ako desiatimi rokmi, kedy bola založená spoločnosť Android Inc. [1] Spoločnosť, ktorá sa zaoberala vývojom operačného systému pre mobilné telefóny. Od jej vzniku bola finančne podporovaná spoločnosťou Google, ktorá ju v roku 2005 odkúpila. 5. novembra 2007 vzniklo konzorcium hardwarových, softwarových a telekomunikačných spoločností Open Handset Alliance, ktorého členom je aj Google. Spolu so vznikom OHA sa uskutočnilo oficiálne predstavenie operačného systému Android. V roku 2008 bol uvedený prvý komerčný telefón s týmto operačným systémom HTC Dream, ktorý predával T-Mobile v USA. V roku 2009 prišli verzie 1.5 Cupcake, 1.6 Donut a 2.0 Eclair. Stúpa podiel na trhu a Android beží na viac ako dvadsiatich zariadeniach od rôznych výrobcov. Nasledovali ďalšie vylepšenia. Verzia 4.0 Ice Cream Sandwich z roku 2011 spojila vetvu 2.x, ktorá slúžila telefónom a 3.x slúžiaca tabletom a vytvorila spoločnú SDK (Software Development Kit - sada nástrojov pre vývoj softwaru na určité platformy).

1.1.2 Vývoj aplikácií

Android umožňuje vývoj aplikácií v programovom jazyku Java. Nevyužíva štandardný virtuálny stroj Java Virtual Machine, ale aplikačný virtuálny stroj Dalvik so súborom inštrukcií na základe registra a svojím vlastným byte kódom.

Programátori majú k dispozícii SDK - Software Development Kit, ktorý poskytuje prakticky všetky prostriedky pre vývoj, teda sadu knižníc, emulátor, ladiace nástroje, dokumentáciu, ukážky zdrojových kódov a návody. Je dostupný pre operačné systémy:

- Windows XP alebo novšie
- Mac OS X vo verzii 10.5.8 alebo novšej
- Linux (Ubuntu od verzie 8.04 Lucid Lynx). Potrebná je knižnica GNU C Library (glibc) verzia 2.7 alebo novšia.

Odporúčané prostredie pre vývoj je Eclipse, podporovaný od verzie 3.6.2 (Helios) spoločne s modulmi Android Development Tools (ADT) a Java Development Tools (JDT) nevyhnutné pre vývoj aplikácií.

Ďalšou možnosťou je využitie akéhokoľvek textového editora, čo je však menej komfortné riešenie.

Súčasťou ADT je emulátor slúžiaci na testovanie aplikácií.

K distribúcii programov slúži Google Play - online distribučná služba spoločnosti Google. Pre distribúciu je potrebné mať platený vývojársky účet spoplatnení sumou 25\$, teda 504 Kč za registráciu. Spoločnosť tvrdí, že tento poplatok je pre zvýšenie kvality produktov v Google Play. [2]

1.2 iOS

iOS je operačný systém firmy Apple určený pre mobilné zariadenia tejto spoločnosti - mobilný telefón iPhone a hudobný prehrávač iPodTouch. Neskôr bola podpora rozšírená na tablet iPad a AppleTV. Spoločnosť Apple neumožňuje použiť iOS na prístrojoch iných výrobcov.

1.2.1 História

Prvý krát bol systém iOS pomenovaný iPhone OS 1.0 uvedený v januári 2007. Nepodporoval aplikácie tretích strán. Miesto toho Apple odporúčal vytvárať webové aplikácie. Na nátlak užívateľov sa to zmenilo vo verzii iPhone 2.0 uvedenej v júli 2008. Bola vydaná prvá beta verzia vývojového balíka - SDK určená pre vývoj natívnych aplikácií. V júni 2004 bol operačný systém premenovaný na iOS spoločne s uvedením verzie 4.0. Tá obsahovala jednu z najvýznamnejších inovácií a to podporu súčasného chodu viacerých aplikácií. Ďalšia verzia iOS 5.0 v októbri 2010 privedla hlasovú asistentku Siri, ktorá nahradila pôvodné hlasové ovládanie.

Vo verzii iOS 6.0 bola nahradená aplikácia Mapy založená na Google Maps aplikáciou s mapovými podkladmi od holandskej firmy TomTom.

V súčasnosti je spoločnosť Apple pred zverejnením iOS 8.0. Beta verzie značia najväčší progres v oblasti komunikácie s operačným systémom na počítačoch a taktiež prepojenie s domácnosťou - HomeKit. Pre vývojárov je dobrou správou povolenie väčšieho zásahu do systému a tak nové možnosti vo vývoji aplikácií.

1.2.2 Vývoj aplikácií

Primárnym jazykom pre vývoj aplikácií pre iOS je objektové C.

Apple ponúka SDK dostupné iba pre operačný systém Mac OS X, ktoré obsahuje nástroje potrebné pri tvorbe aplikácií. Vývojové prostredie pre vytváranie aplikácií X Code, vývojové

prostredie umožňujúce webové aplikácie pre iPhone a iPad Dashcode, emulátor iPhone a iPad, nástroje na ladenie výkonu a hľadanie chýb pri vývoji aplikácií. iOS knižnica pre vývojárov je dostupná aj na stránkach Apple Developer. Obsahuje dokumentáciu, zdrojové kódy, množstvo návodov a rád, ako postupovať pri vývoji aplikácií. iOS je pomerne uzavretý systém. K distribúcií slúži výhradne oficiálny Apple App Store, obsahujúci stovky tisíc aplikácií pre iPhone a iPad. Pre publikáciu aplikácie do App Store je potrebný platený vývojársky účet a certifikát k podpisu aplikácie. Tá musí pred sprístupnením prejsť niekoľkodňovým schvaľovacím procesom. Ročná cena účtu pre vývojára je 99\$ teda 1995 Kč. [3]

1.3 Microsoft Windows Phone

Windows Phone je operačný systém pre mobilné telefóny vyvinuté firmou Microsoft. Je licencovaný pre zariadenia rôznych výrobcov, ale najviac zariadení s touto platformou predáva spoločnosť Nokia. Windows Phone definuje striktné hardwarové požiadavky, ktoré musí zariadenie splniť, aby bolo pre tento operačný systém použiteľné.

1.3.1 História

Prvá verzia nesúca označenie Windows Phone 7 bola uvedená v októbri 2010. Jeho predchodcom bol operačný systém Windows Mobile. Bol založený na jadre Windows CE. Nasledovala verzia Windows Phone 7.5 Mango, vyšla v máji 2011. Priniesla vylepšenia ako multitasking pre aplikácie tretích strán, hlbšiu integráciu sociálnych sietí Twitter a LinkedIn do systému a nová verzia internetového prehliadača Internet Explorer 9. V októbri 2012 vyšla verzia Windows Phone 8 Apollo. Systém bol kompletne prepracovaný tak, že využíva jadro Windows NT, rovnako ako Windows 8. Získal tým rovnaký súborový systém, ovládače zariadení, časti bezpečnostného modelu, grafický motor a hardwarovú abstraktnú vrstvu.

Vývojári tým získali možnosť ľahkého prenášania aplikácie medzi týmito dvoma platformami.

1.3.2 Vývoj aplikácií

Vývoj aplikácie pre platformu Windows Phone 7 je možný dvomi spôsobmi. Je možné použiť technológiu Microsoft Silverlight vhodnú pre vývoj bežných aplikácií alebo Microsoft XNA, ktorá umožňuje najmä vytvárať 2D a 3D hry s vysokým grafickým výkonom.

Silverlight technológie sú písané v jazyku C alebo Visual Basic. Grafické užívateľské rozhranie aplikácie sa definuje pomocou značkovacieho jazyka XAML.

Pre Windows Phone 8 bol uvedený nový programovací model a kolekcií rozhraní Windows Runtime. Pre vývoj aplikácií pre Windows Phone 8 je nutné nainštalovať Windows

Phone 8 SDK obsahujúci okrem iného aj knižnice, projektové šablóny a emulátor založený na virtuálnej technológii Windows Hyper-V.

Aplikácie sú distribuované prostredníctvom Windows Store. Pre uverejnenie aplikácií je požadovaný platený vývojársky účet, pričom publikovaná aplikácia musí prejsť niekoľkodňovým schvaľovacím procesom, pri ktorom sa kontroluje okrem iného aj striktné dodržiavanie oficiálnych pravidiel pre vzhľad užívateľského rozhrania. Pre českých vývojárov je stanovená cena za užívateľský účet 365 Kč ročne. [4]

1.4 Ostatné mobilné platformy

1.4.1 Symbian

Symbian je voľne šíriteľný operačný systém navrhnutý pre mobilné zariadenia. Je udržiavaný spoločnosťou Accenture, ktorej to prenechala spoločnosť Nokia. Bol najrozšírenejší operačný systém pre múdre telefóny do roku 2010, kedy ho vystriedal Android. V roku 2011 bolo oznámené partnerstvo Nokie so spoločnosťou Microsoft a prechod na platformu Windows Phone. Spoločne s prechodom spoločnosti Motorola, Samsung, LG a Sony Ericsson na platformy Android Bada to spôsobilo odchod vývojárov od platformy Symbian. Aktuálna verzia systému sa volá Nokia Belle FP2.

Pre vývoj aplikácií sa používa programový jazyk C++ s použitím Qt SDK. Obsahuje dokumentáciu, knižnice, hlavičky súborov, kompilátor a emulátor prístroja.

1.4.2 Samsung Bada

Bada je platforma firmy Samsung vyvíjaná pre jej smartfóny. Ohlásená bola v roku 2009 a na trh uvedená o rok neskôr. Doplnením o dotykové ovládanie, 3D grafiku, prepracovaním grafického rozhrania, doplnením možnosti sťahovania aplikácie tretích strán sa z neho stal plnohodnotný operačný systém pre mobilné prístroje.

Operačný systém Bada môže používať jadro Linuxu alebo iné RIOS jadro (jadro operačného systému reálneho času). Nad jadrom sú vrstvy zariadenia, služieb a softvérový rámec. Pri vývoji sú k dispozícii dva rámce, C++ použiteľný na vývoj flash alebo C++ aplikácií a webový rámec používaný na vývoj webových aplikácií pomocou technológií ako HTML, CSS a JavaScript.

Pre vývojárov vydal Samsung SDK, ktoré sa dá voľne stiahnuť.

Pre vytvorenie novej aplikácie je nutné zaregistrovať sa na vývojárskych stránkach Bada a vytvoriť nový profil pre aplikáciu. Podľa nastavení vlastností aplikácie sa vygeneruje súbor manifest.xml, ktorý je nutné stiahnuť a vložiť do nového projektu aplikácie. Ak je hotová aplikácia schválená, je zaradená do predaja v internetovom Samsung App.

1.4.3 **Blackberry**

BlackBerry je proprietárny mobilný operačný systém, vyvinutý kanadskou firmou Research In Motion pre používanie v ich smartfónoch BlackBerry. Systém podporuje multitasking a viacero špecializovaných vstupných zariadení, prispôsobených firmou RIM pre použitie s ich zariadeniami. Hlavnou výhodou tejto platformy je natívna podpora podnikového e-mailu cez aplikačné rozhranie MIDP. MIDP podporuje kompletnú bezdrôtovú aktiváciu a synchronizáciu e-mailov, úloh, kalendára, poznámok a kontaktov cez Microsoft Exchange, IBM Lotus Domino alebo Novell GroupWise za podmienok, že sa používajú spoločne so serverom BlackBerry Enterprise Server.

2 Multiplatformové aplikácie verzus natívne aplikácie

2.1 Výber architektúry

Optimálny výber mobilnej aplikácie závisí od druhu vyvíjanej aplikácie, jej požiadavkách, na cieľovej skupine užívateľov a na perspektíve jej ďalšieho rozvoja. Výber vhodného prístupu je pre vývoj aplikácie zásadný, pretože ovplyvňuje jej ďalšiu implementáciu.

Jedná sa o prístup aplikácie:

- natívny, ktorý je vhodný na mobilné hry, prehrávanie hudby a videa. Má dobrú ovládateľnosť a rýchlu odozvu, vysoký grafický výkon a možnosť prístupu aplikácie na internet.
- webový je vhodný pre aplikácie, ktoré ponúkajú služby zákazníkom, pre internetový obchod alebo sociálne siete, ktoré slúžia na komunikáciu s užívateľmi. Je závislý na internetovom pripojení a nenáročný na ovládanie.
- hybridný - multiplatformový prístup pozostáva zo spojenia funkcie natívneho a webového prístupu a ich vzájomného využívania. Je podporovaný všetkými hlavnými mobilnými platformami.

2.1.1 Natívne aplikácie

Natívna aplikácia je riešená vždy na mieru pre jednu konkrétnu platformu a prostredníctvom API operačného systému má prístup ku hardwarovým funkciám zariadenia, (napr. GPS, akcelerometer, gyroskop, kamera a pod.) ktoré sú prístupné cez natívne rozhranie zariadenia. Môže využívať pokročilé funkcie zariadenia, pristupovať k dátam z rôznych senzorov a používať spoločné komponenty systému, ktoré sú k dispozícii.

Táto aplikácia je schopná fungovať aj bez pripojenia na internet (dáta je možné uložiť do pamäte zariadenia a následne k nim pristupovať). Pri natívnej aplikácii je potrebné pridávať podporu pre nové zariadenia, ktoré sa objavujú na trhu, pridávať pravidelne aktualizácie a zverejňovať ich, čo zvyšuje náklady na správu.

Nevýhodou natívnej aplikácie je skutočnosť, že je vždy funkčná len na jednej konkrétnej platforme (natívnu aplikáciu kompilovanú pre iPhone nie je možné použiť pre Android apod.).

Pre ďalší vývoj techniky je natívna aplikácia pre mobilné telefóny neperspektívna, pretože aplikácie spolu s platformami rýchle starnú.

2.1.2 Webové aplikácie

Webové aplikácie sú nástroje, ktoré umožňujú vyvíjať aplikácie pre viacero platforiem súčasne a tým podporujú ich distribúciu. Výhodou tohto prístupu je využitie už existujúcej

webovej aplikácie, na ktorej je možné vytvárať nové mobilné aplikácie. V súčasnosti všetky mobilné zariadenia disponujú webovým prehliadačom.

Webová aplikácia existuje na jednej inštalácii, je spravovaná na jednom mieste, čo uľahčuje inštaláciu, aktualizáciu, konfiguráciu apod., čím sa zaručí používanie najnovšej verzie aplikácie. Webová aplikácia pracuje na serveri a má zjednodušený prístup k serverovým zdrojom a databázam. Webové aplikácie sú vyvíjané pomocou technológií, ako PHP, HTML5, CSS a JavaScript a sú umiestnené na webových serveroch.

Nevýhodou webovej aplikácie je nutnosť prístupu na internet, čo môže byť pre mnohé aplikácie nestabilné, pretože mobilné zariadenia sa často pohybujú v oblastiach so slabým signálom, čo je pre užívateľov nepohodlné.

2.1.3 Multiplatformové - hybridné aplikácie

Vznik multiplatformových - hybridných aplikácií bol vynútený vývojom nových platforiem na trhu, ich fragmentáciou, prispôbovaním sa a potrebou efektívne riadiť ich vývoj. Môžu byť zamerané na mobilné platformy (napr. Apple iOS, Google Android, Microsoft Windows Phone 8), tablety, desktopy (Microsoft Windows, Apple OSX, Linux) a herné konzoly (Sony PlayStation, Microsoft Xbox360).

Hybridný prístup viac menej kombinuje výhody a nevýhody natívnej a webovej aplikácie. Hybridné aplikácie sa skladajú z natívnej schránky (wrapper), ktorá obsahuje HTML, CSS a JavaScript kód. Ak je táto aplikácia na zariadení spustená, wrapper spustí inštaláciu webového prehliadača (WebView) a nahrá HTML, CSS a JavaScript. Komponent WebView nemá ovládacie prvky webového prehliadača, čím aplikácia vyzerá rovnako ako natívna.

Pre užívateľov sú hybridné aplikácie úplne rovnaké ako natívne. Sťahujú sa z obchodov natívnych platforiem a inštalujú rovnako. Pre vývojára ich vývoj pripomína vytváranie natívnych aplikácií tým, že je možné použiť HTML, CSS a JavaScript kód na napísanie aplikácie. Navyše hybridný model je podporovaný všetkými hlavnými mobilnými platformami, čo umožňuje opätovné použitie už napísaného kódu.

Hybridné riešenie je vhodné aj v situáciách, keď je potrebné v aplikácii pracovať s dátami, ktoré sú prístupné iba online cez existujúce webové rozhranie a ostatné dáta je možné doručiť natívnym spôsobom.

Hlavné výhody vieme charakterizovať v týchto bodoch:

- možnosť vytvárať a zverejňovať aplikácie na rôzne platformy
- minimalizácia nákladov na jednu platformu
- možnosť opätovného použitia kódu na rôznych platformách
- nie sú obmedzené len podporou mobilných platforiem, ale umožňujú vývoj aj pre počítačové systémy a herné konzoly

- úspora času na zverejnenie aplikácie do obehu
- úspora finančných prostriedkov pre vývojársku spoločnosť

Nedostatky multiplatformových aplikácií:

- znížený výkon v porovnaní s natívnym alebo webovým riešením, čo je spôsobené neustálym prechodom medzi natívnymi a webovými komponentmi. Je potrebná JavaScript abstrakcia kvôli zmenšovaniu rozdielov medzi natívnymi rozhraniami
- pre potrebu rýchlych reakcií aplikácie nie je hybridný prístup príliš vhodný

3 Multiplatformové softvérové rámce

3.1 PhoneGap

PhoneGap spoločnosti Adobe bol vyprodukovaný medzi prvými hybridnými aplikáciami a zároveň pod voľne šíriteľnou licenciou sa stal súčasťou mnohých multiplatformových nástrojov. Je najrozšírenejší nástroj na tvorbu mobilných aplikácií.

PhoneGap predstavuje obaľovače webového kódu, ktoré umožňujú pristupovať k natívnym funkciám zariadenia a pritom stále bežať vo vnútri prehliadača. Jadro aplikácií tvorených nástrojom PhoneGap je písané v HTML, CSS a JavaScripte. Vyvíja sa na štandardných editoroch, ako Dreamweaver alebo Eclipse.

PhoneGap patrí medzi obaľovače, to znamená že pri tvorbe aplikácie zapuzdruje webový kód spolu s vlastnými knižnicami do natívnej spustiteľnej schránky. Pri vytváraní aplikácií sa využívajú aj API rozšírenia, ktoré umožňujú pristupovať k hardwarovým funkciám zariadenia, ako sú akcelerometer, gyroskop a GPS.

PhoneGap sa zameriava na webových vývojárov, ktorí chcú vyvíjať natívne aplikácie a následne ich distribuovať cez natívne trhy. Pokrýva všetky veľké mobilné platformy ako iOS, Android, BlackBerry, Bada, Windows, Mobile Phone 8 a Symbian. Hlavným dôvodom jeho vzniku bol zámer poskytnúť vývojárom mobilných webových aplikácií prístup k hardwarovým funkciám prístroja bez nutnosti učiť sa programovacie jazyky, v ktorých sú natívne aplikácie písané. PhoneGap tiež ponúka službu nazvanú PhoneGap Build, ktorá umožňuje nahráť svoju aplikáciu napísanú pomocou jazykov HTML, CSS a JavaScript na server, kde sa spracuje do natívnej aplikácie pre zvolenú platformu. Vďaka tomu nemusí vývojár inštalovať SDK na svoj počítač a samostatne kompilovať aplikáciu pre každú platformu.

PhoneGap v nedávnej minulosti často pletený s Apache Cordovou. Nie je na to dôvod. PhoneGap je distribúcia Apache Cordova, to znamená, že Cordova je ako keby motor PhoneGapu, ktorý ho poháňa. Jeden od druhého sa teda vôbec nelíšia a oba ostávajú voľne šíriteľné s otvorenou licenciou. Rozdiel je jedine v tom, že zatiaľ čo PhoneGap je určený pre konečných vývojárov, Apache Cordova ako tiež voľne šíriaci projekt s otvorenou licenciou je možné aj pomáhať rozvíjať.

3.2 Marmalade

Marmelade SDK je multiplatformový nástroj vyvinutý spoločnosťou Ideaworks3D. Obsahuje knižnice, návody, dokumentáciu a nástroje potrebné na testovanie a vývoj aplikácií pre mobilné zariadenia. Vývoj je možný na operačných systémoch Windows alebo OS X a prebieha v jazyku C++ za pomoci multiplatformových rozhraní Marmelade.

Marmelade je definovaný ako prekladač zdrojového kódu do medzikódu, natívneho jazyka alebo priamo do nízkoúrovňového strojového kódu. Princípom prekladača je to,

že funkčná časť aplikácie je naimplementovaná v niektorom rozšírenom jazyku a pre každú platformu zvlášť je vytvorené užívateľské rozhranie. Je to výhoda, nakoľko užívateľské rozhranie je u jednotlivých operačných systémoch rôzne, zatiaľ čo aplikačná logika je pre všetky platformy rovnaká.

Spoločnosť Ideaworks3D sa najskôr zameriavala na herných vývojárov pre prácu s 3D grafikou a animáciami priamo v SDK, neskôr bola pridaná podpora aj pre natívne prvky užívateľských rozhraní jednotlivých platforiem, čím sa rozšírila cieľová skupina zákazníkov aj o neherných vývojárov (cca 80% užívateľov Marmelade ho využíva na tvorbu hier).

3.3 Appcelerator Titanium

Appcelerator s názvom Titanium bol vyvinutý v roku 2008 spoločnosťou Appcelerator Inc. Aplikácie tohto nástroja sú písané v jazykoch HTML, CSS a JavaScript s podporou PHP, Ruby a Python pre počítačové aplikácie. Na prístup k natívnym funkciám zariadenia sa používa programové rozhranie Appcelerator API.

Titanium premoštuje JavaScript na natívne knižnice a vykoná kompiláciu do bytekódu. Následne kompilátor danej platformy vytvorí inštalovateľný balíček a výsledný binárny súbor obsahuje väčšinou natívny kód spolu s interpreterom jazyka JavaScript a renderuje sa natívne. Takto vytvorené aplikácie nebežia vo vstavanom webovom pohľade, ale využívajú len samotné JavaScript jadro, pomocou ktorého je interpretovaný kód a zabezpečený prístup k systémovým funkciám prístroja.

Titanium Mobile je vývojový rámec, ktorý umožňuje vyvíjať aplikácie pre platformy iOS, Android a BlackBerry. Vývoj pre Android a BlackBerry vyžaduje Android SDK, resp. BlackBerry SDK a môže byť vykonávaný na operačných systémoch Windows, Linux a OS X.

Titanium poskytuje prístup k hlavným hardvérovým a systémovým funkciám prístrojov, ako napr. GPS, fotoaparát, zoznam kontaktov, prehrávanie video a audio záznamov a mnoho ďalších. Titanium SDK je dostupné pod voľne šíriteľnou licenciou Apache 2.

3.4 Ostatné softvérové rámce

Softvérové rámce zažívajú obrovský rozkvet a z môjho pohľadu vývin v tejto kategórii aplikácií zažije veľkú revolúciu. Mnou spomenuté tri softvérové rámce sú v tejto dobe prenasledované veľkým množstvom konkurentov. Za zmienku určite stojí RhoMobile, Sencha, Adobe AIR, Qt či ďalšie. Očakávam veľkú zmenu, pretože podobne ako spoločnosť Adobe Systems kúpila v roku 2011, môže nastať veľká investícia do ktoréhokoľvek z uvedených, či dokonca ešte softwarových rámcov, ktoré sú vo vývoji a nemajú ani jedného užívateľa.

Skvelým pomocníkom pri výbere správneho softvérového rámca je webová stránka www.mobile-frameworks-comparison-chart.com. Už na prvý pohľad registrujeme, aký rozkvet

tento sektor informatiky zažíva a aké nekonečné možnosti máme pri výbere softvérového rámca. Na tejto stránke si môžeme veľmi jednoducho zadať naše potreby pre nastávajúci projekt a systém nám automaticky ponúkne tie ideálne. Rozhodovanie ja tak prehľadné a oveľa jednoduchšie.

3.5 Porovnanie a hodnotenie

Multiplatformové nástroje a ich úspech má hlavnú príčinu. Podľa prieskumov z dvoch tretín je príčinou pre výber týchto nástrojov práve dostupnosť všetkých softvérových rámcov. Minimalizácia nákladov je hlavne pri nových projektoch, či menších firmách tým hlavným kritériom. Ďalšími dôvodmi sú rýchlosť vývoja a krivka vzdelanosti v danom jazyku.

Najpoužívanejším nástrojom v tejto dobe je PhoneGap. Jednoznačne za tým stojí aj silný majiteľ Adobe Systems, ktorý pomáha dopomáhať vývojárom k zdarným projektom či už s návodmi, alebo centrom podpory. Aj nadstavba na PhoneGap, PhoneGap Build, ktorá dokáže veľmi jednoducho vytvoriť aplikáciu aj pre úplných začiatočníkov dáva tomuto nástroju krídla. Podľa výsledkov z roku 2013 až 34% vývojárov verí tomuto nástroju.

PhoneGap však nie je jediný úspešný a vývojári často siahajú aj na kombináciu dvoch, či viacerých nástrojov. Na zhodnom druhom mieste sa umiestnili Appcelerator a Adobe AIR, ktoré majú jednu pätinu trhu. O kombinácii svedčí aj fakt, že pri jednej aplikácii je priemerne využitých 1,91 multiplatformových softvérových rámcov.

4 Návrh multiplatformovej aplikácie v softvérovom rámci PhoneGap

4.1 Predstavenie aplikácie

Aplikácia je navrhovaná pre miestnu ZOO v Ostrave. Vďaka dômyselnému jednoduchému užívateľskému rozhraniu sa bude môcť každý účastník v ZOO jednoducho zorientovať. V aplikácii bude zahrnutá mapa zvierat, kde sa bude môcť jedným klikom dozvedieť potrebné informácie o danom zvierati. Taktiež budú v aplikácii zahrnuté všetky informácie, ktoré zvýšia komfort, informovanosť a zlepšia orientáciu návštevníkov v ZOO. Príkladom môže byť mapa všetkých toaliet, navigácia užívateľa z miesta, v ktorom sa nachádza do miesta prvej pomoci. Všetky tieto vylepšenia bližšie popíšem v nasledujúcich podkapitolách. V aplikácii budú zahrnuté osobitne všetky dôležité informácie, ktoré sa ZOO týkajú. Tieto dôležité informácie budú obsahom hlavného menu z dôvodu lepšej informovanosti návštevníkov ohľadom ZOO. Príkladom môže byť adopcia zvierat'a, čo znamená, že každý návštevník bude môcť prispievať na stravu vybraného zvierat'a.

4.2 Použité technológie a softvérové rámce

V krátkosti uvediem, ktorý multiplatformový softwarový rámec je využitý a v krátkosti uvediem dôvody výberu. Okrem technických parametrov, ktoré so sebou prináša tvorba aplikácie pre ZOO boli zvažované moje predchádzajúce skúsenosti. Ďalším dôležitým faktorom výberu softwarových bolo udelenie licencie k jeho používaniu.

4.3 Použité zásuvné moduly

Vo svojej bakalárskej práci využívam rôzne zásuvné moduly, ktoré sa implantujú do multiplatformového softwarového rámca PhoneGap. V nasledujúcich podkapitolách v krátkosti popíšem, ktoré sú v práci použité. Taktiež je popísané, kde sa presne v aplikácii využili.

4.3.1 Google Android Maps API V2

Zásuvný modul Google Maps API V2 je použitý hlavne z dôvodu natívnosti. V tejto bakalárskej práci bolo možné poznať rozdiely medzi natívnym a webovým prístupom a pri prezentovaní Google Android Maps API V2 oproti Google Maps API V3. Natívne aplikácie nám ponúkajú možnosti, ktoré webové ešte nedokážu. Práca v natívnej aplikácii je príjemnejšia, odozva je rýchlejšia a ponúka aj väčšie možnosti.

Zobrazenie natívnej mapy vidíme v prvej položke hlavného menu "Chodníčky v našej ZOO". Vidíme popísané každé zviera, ktoré sa v ZOO nachádza. Presne nasmerovanie na mape pomocou Google Maps Polyline nás navedie presne tam, kde to budeme potrebovať, aby sme žiadnu dôležitú cestu v ZOO neminuli.

Zásuvný modul je použitý pri funkcii "Vieď mňa do ZOO". Po otvorení tejto záložky sa zobrazí trasa z miesta, v ktorom užívateľ stojí do miesta, kde je ZOO situovaná. Taktiež je využitá táto verzia pre rýchle zobrazenie trasy k prvej pomoci, aby užívateľ sa nemusel starať o zorientovanie sa na mape a iba nasledovať inštrukcie od aplikácie.

4.3.2 Google Maps API V3

Google Maps API V3 pomáha pri vytváraní Google máp do mobilných aplikácií a internetových stránok, vďaka množstvu návodov, ktoré sú dokonca preložené do slovenčiny alebo češtiny. Pre tieto dôvody je tento zásuvný modul použitý v bakalárskej práci.

Google Maps API V3 bol vybraný aj z dôvodu jediného obmedzenia, ktoré sa vzťahovalo na počet využívania máp za jeden deň. Počet využitia máp Google stanovil na 25 000, čo znamená, že mapu si môže rozkliknúť 2 000 užívateľov za deň, čo je v našom prípade plne postačujúce.

Zásuvný modul Google Maps API V3 je použitý pri zobrazovaní mapy zvierat. Užívateľ vďaka jednoduchému otvoreniu ikony Mapa zvierat uvidí, v ktorej časti ZOO sa nachádza ktoré zviera. Pomocou prídavného modulu od Googlu som pridal vlastné ikonky, viď Obrázek 1.1 Ikonky v mapách aplikácie.



Obrázek 1.1: Ikonky v mapách aplikácie

Vlastná značka sa do aplikácie pridá kódom, viď Kód 1.1. Pridanie značky do mapy. Každý užívateľ sa po otvorení ikony dostane do sekcie, ktorá popisuje dané zviera. Vo vybranom popise je názov zvierat a vybrané dôležité informácie. Pri každom popise zvierat sa nachádza mapa vo forme zobrazenia ulíc, ktorá určuje presnú polohu. Užívateľ sa tak môže presne zorientovať, kde sa zviera nachádza. Pridanie zobrazovania ulíc sa vďaka zásuvnému modulu Google Maps API V3 pridá nasledovným spôsobom, viď Kód 1.1 Pridanie značky do mapy.

```
$('#map_canvas').gmap('addMarker',{'position': 49.847814,  
18.328037, 'bounds': true}).click(function() {  
  
$('#map_canvas').gmap('openInfoWindow', {'content': 'Restarurace  
Saola'}, this); }
```

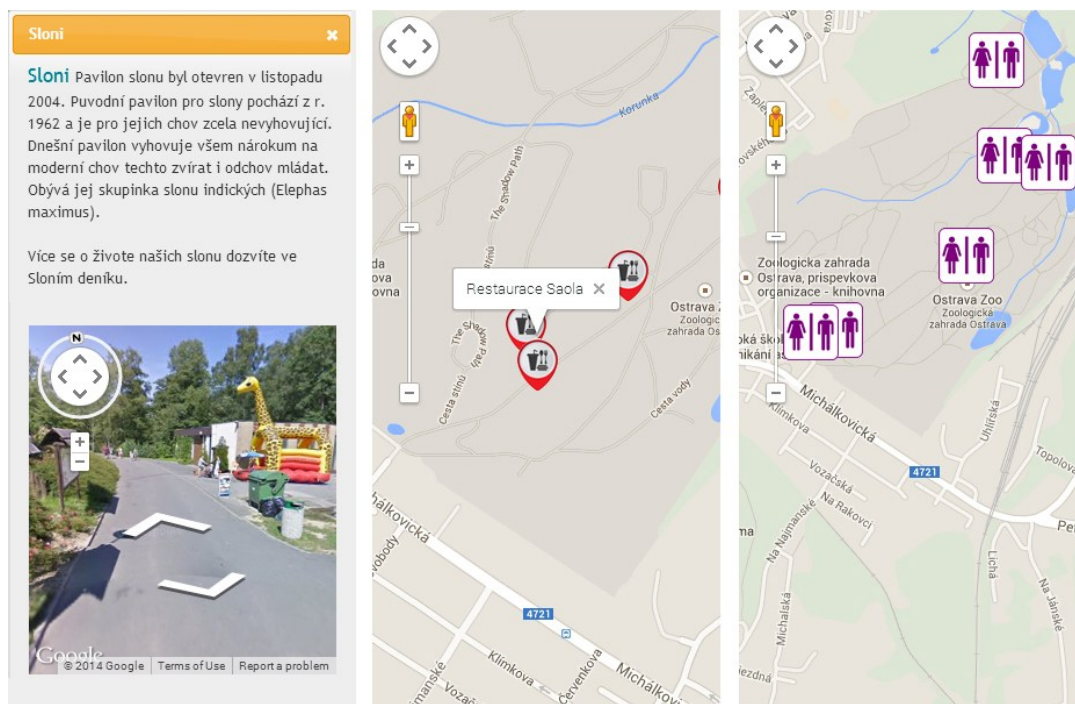
Kód 1.1: Pridanie značky do mapy

```
self.displayStreetView('streetview{0}'.replace('{0}', index), {
'position': $(this)[0].getPosition() });
```

Kód 1.2: Pridanie zobrazovanie

Taktiež bol tento zásuvný modul použitý pri zobrazovaní všetkých toaliet či občerstvenia. Po otvorení ikonky všetkých toaliet, ktoré sa v ZOO nachádzajú a tak si bude môcť užívateľ zvoliť pre neho vhodnú toaletu, viď Obrázek 1.2 Využitie Google Maps API V3.

Ikonka jedlo a pitie slúži na zobrazenie všetkých občerstvovaniach staníc, ktoré sa v ZOO nachádzajú. Mapa zobrazuje ikonky, ktoré po otvorení zobrazia názov občerstvovanej stanice.



Obrázek 1.2: Využitie Google Maps API V3

4.3.3 Push notifikácia

Prídavný modul PushPlugin slúži na posielanie notifikácií ohľadne udalostí, ktoré sa stali alebo práve prebiehajú. Ak sa užívateľ nachádza v ZOO Ostrava, pomocou notifikácií bude informovaný v reálnom čase o aktuálnych novinkách, informáciách, či podujatiach, ktoré by nemal zmeškať.

Po značnom vývoji sa však tento prídavný modul nepodaril implementovať. Vzniknutá situácia má viacero dôvodov. Tým hlavným je možnosť zárobku firiem, ktorý podobný spôsob

notifikácie vyvinú, snažia sa ho predať ďalej. Veľmi náročný vývoj natívnych aplikácií, ktoré využívajú geolokáciu, či časové zóny pre presnejšiu špecifikáciu zákazníkov je veľmi náročné na čas. Pri tejto bakalárskej práci som sa stretol s veľkou ponukou prídavných modulov a vedel by som niektoré z nich využiť presne podľa mojich potrieb. Všetky tieto spoločnosti však ponúkali iba možnosť zaplatenia ich služby. Jedinou možnou spoločnosťou s voľne šírením prídavným modulom bola spoločnosť Infobip.

Implementácia notifikácií sa mi podarila na testovacej aplikácii a následne bola prevedená implementácia na aplikáciu ZOO Ostrava. V tom čase som však zistil zmenu účtu a hneď mi bola náúčtovaná suma, za užívanie tejto služby. Aj napriek tomu, že spoločnosť avizuje 1 000 000 voľných notifikácií aj s využitím geolokácie.

Zistenie, že vývoj tejto sféry natívnych prvkov je spoplatnený na všetkých úrovniach nás informuje, že daná sféra stojí veľké investície pri vývoji. Voľne šíriteľné prídavné moduly nespĺňujú potrebné možnosti, hlavne čo sa týka informačného systému. Je totiž nutný následný vývoj vlastného informačného systému, ideálne na systéme webového servera, ktorý bude notifikácie obstarávať.

4.3.4 **jQuery Mobile**

jQuery Mobile je HTML5 orientované užívateľské rozhranie, ktoré nám vytvára responzívne webové stránky a aplikácie, ktoré sú spustiteľné na všetkých mobilných aplikáciách, tabletoch a počítačoch.

Tento prídavný modul nám umožňuje pracovať s HTML5, CSS3, JavaScriptom, či Ajaxom veľmi jednoducho. Takzvaná pís menej, urob viac metóda jQuery Mobile je postavená na knižnici jQuery a tak už teraz je veľké množstvo vývojárov schopné s týmto prídavným modulom pracovať.

4.4 **Ciele aplikácie**

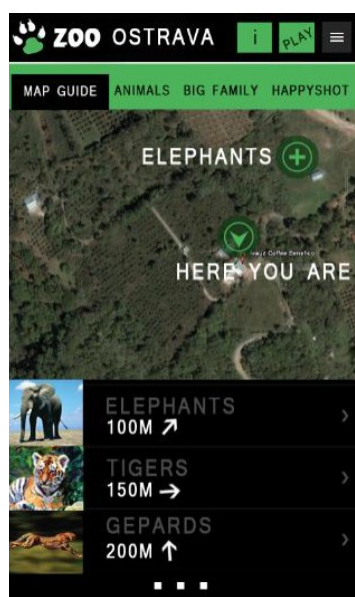
Cieľom vytvorenia aplikácie bolo prakticky preskúmať možnosti webových technológií ako HTML5, CSS3 a JavaScript. Pomocou zmienených technológií vytvoriť geolokačnú aplikáciu, ktorá bude schopná fungovať na rôznych mobilných a webových platformách. Aplikácia je určená pre miestnu ZOO v Ostrave.

5 Implementácia aplikácie Mobilný sprievodca v ZOO

5.1 Prvotný návrh

Na Obrázku 1.3: Návrh aplikácie môžeme vidieť prvotný návrh aplikácie. Po zadaní bakalárskej práce boli moje vedomosti v oblasti multiplatformových aplikácií na minimálnej úrovni. Nový druh aplikácií do mobilných telefónov mi ukázal cestu, ktorou si myslím že sa vydá hlavná časť vývojárov. Veľmi jednoduché doškolenie webových vývojárov, ktorí teraz majú možnosť vyvíjať aj mobilné aplikácie a veľmi rýchly vývin týchto softwarových rámcov mi dáva za pravdu.

V prvotnej fáze projektu boli moje vedomosti o multiplatformových aplikáciách minimálne. Nedostačujúce informácie, aké sú možnosti, aké prednosti a kde sú zase nevýhody. Využitie ale boli skúsenosti s programovaním Android aplikácie. Návrh sa líši od finálnej verzie hlavne kvôli obmedzeniam, ktoré multiplatformové aplikácie majú. Využívanie prídavných modulov napríklad od Google je tiež veľmi obmedzené. Pre tieto skutočnosti je finálna aplikácia rozdielna a ako základ uvítacej stránky tvorí hlavné menu.



Obrázek 1.3: *Návrh aplikácie*

5.2 Vývoj aplikácie

Ak chceme vytvoriť aplikáciu, ktorá má byť veľmi jednoduchá pre malé deti, ako aj starých rodičov, musíme pracovať veľmi opatrne. Výber správneho nástroja pre tvorbu je veľmi dôležitý a preto výber práve multiplatformovej aplikácie je správna voľba.

Pre nenáročnosť aplikácie vieme veľmi jednoducho eliminovať najväčšie nevýhody multiplatformových aplikácií a to je hlavne ich rýchlosť. Pri ďalších problémoch, ktoré v tomto segmente informatiky nachádzame nenájdeme až také problémy, keďže aplikácia informačného charakteru si postačí aj s tým základným využitím natívnych schopností zariadenia. Pri výbere toho správneho nástroja vieme veľmi jednoducho nájsť to pravé a začať s tvorbou.

5.3 Odporúčaný vývoj aplikácie

Vypracovanie bakalárske práce bolo riadené základnými princípmi vývoja či už webovej či mobilnej aplikácie alebo aplikácie pre počítače. Základné princípy boli potvrdené a mojím prísunom vidím odporúčanie, na ktoré časti sa v prípade multiplatformových aplikáciách zamerať viac a na ktoré časti sa zamerať menej.

5.3.1 Analýza

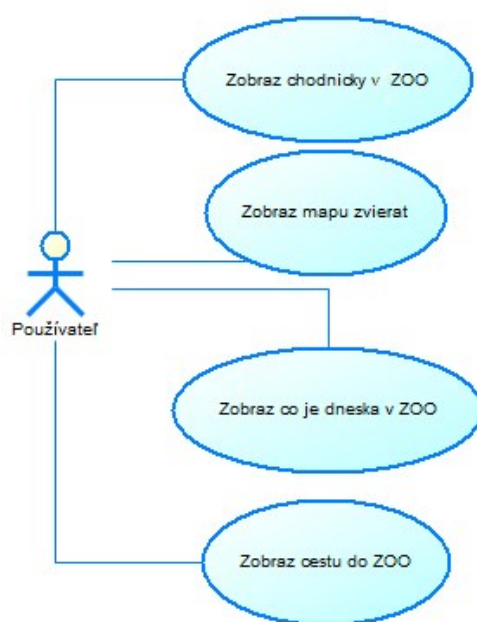
Výber toho správneho nástroja je základom úspechu. Keďže veľmi veľa firiem, ktoré vyvíjajú tieto platformy sú stále vo vývoji, je dôležitým faktorom aj kvalita, či kvantita. V prípade kvality je bezpochyby najlepším príkladom spolupráca Adobe Systems s PhoneGapom, alebo pomocou Adobe AIR. Veľmi kvalitne prevedené návody, dokumentácia či flexibilné pomocné užívateľské centrum. V prípade kvantity by som spomenul neodmysliteľné testovanie, vyriešenie všetkých problémov, či porúch nástroja.

Len veľmi málo vývojárov má však čas na to, aby si mohli dovoliť zvoliť istý vývojársky nástroj a po vzniknutí komplikácií ho vymeniť za druhý a začať s prácou od začiatku. Preto odporúčam neprehliadnuť túto časť tvorby a aj jej vážnosť. [5]

5.3.2 Vývoj

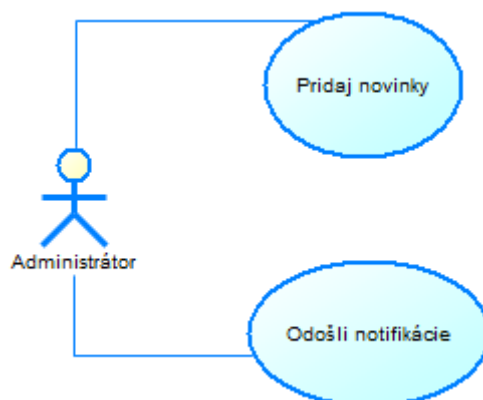
Po zvolení prvého nástroja sa dostávame do fáze prvých aplikácií. Jednoduché textové súbory nahrádzajú prvé využívanie natívnych prvkov. Každé prevedenie aplikácie na pripojené zariadenie alebo nainštalovaný emulátor nám ukáže pokroky aplikácie, ako aj stále pribúdajúce chyby. Na konci vývoja však máme užívateľské rozhranie, ktoré viac či menej sa zhoduje s návrhom. Často zatiaľ nefunkčné časti však musia nadobúdať zmysel a vývojár musí mať jasnú predstavu, aký bude ďalší postup softwarového riešenia.

Pre ukážku prípadov boli vybrané 3 diagramy. Na prvom obrázku, vid' Obrázek 1.4 Využitie API máp, sú zobrazené prípady použitia, aké má k dispozícii užívateľ v hlavnom menu aplikácie.



Obrázek 1.4: *Využitie API máp*

Na ďalšom diagrame, vid' Obrázek 1.5 Administrátor aktualizuje, sú zachytené prípady použitia, ktoré môže vykonávať administrátor aplikácie cez službu Google Cloud Messaging.

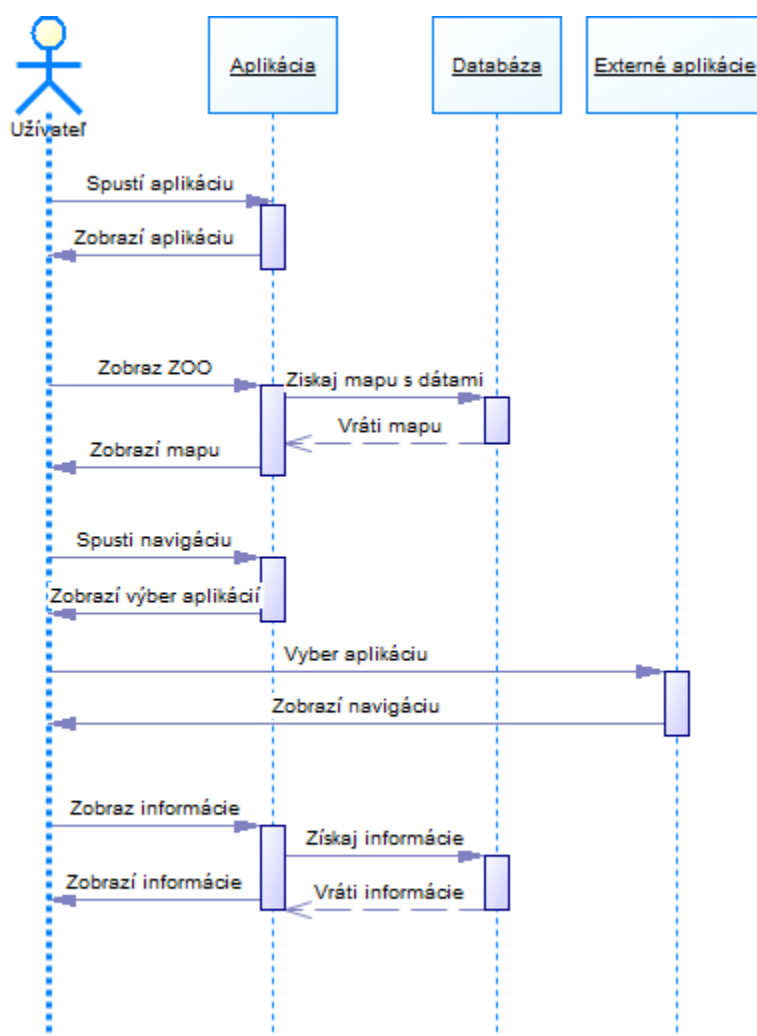


Obrázek 1.5: *Administrátor aktualizuje*

Takisto sú zachytené ešte prípady použitia užívateľa, ktoré si môže zvoliť vo vedľajšom menu, vid' Príloha A Využitie informácií v prílohe.

Po dokončení užívateľského rozhrania sa dostávame do fázy prepojenia grafiky s funkčnosťou. Vďaka prídavným rámcom či návodom si vieme pomôcť rýchlosťou tvorby. Stratíme však dôležitú jedinečnosť a s tým spojenú detailnú úpravu presne podľa svojej potreby, či potreby ďalšieho vývoja aplikácie pre užívateľa.

Na sekvenčnom diagrame, viď. Obrázek 1.6 Sekvenčný diagram, sú zachytené postupy, ktoré sa dejú v aplikácii pri interakcii s užívateľom. Môžeme na ňom vidieť napríklad sekvenciu správ, ktoré sú posielané medzi objektmi a užívateľom pri spustení aplikácie, zobrazení ZOO. Takisto tu môžeme vidieť postup pri zobrazovaní navigácie a zobrazovaní informácií rôzneho druhu, ako sú napríklad otváracia doba ZOO alebo vstupné.



Obrázek 1.6: *Sekvenčný diagram*

Pri dokončení tejto časti získame aplikáciu, ktorá je schopná pomôcť užívateľovi, je však veľmi nepraktická pre klienta a tým je v našom prípade ZOO Ostrava.

5.3.3 Pripojenie

Je nepostačujúce vyrobiť aplikáciu, ktorú klient nemôže za okamžik upraviť, prípadne by musel využiť služby programátora. Aplikácia mobilný sprievodca v ZOO je aktualizovaná od ľubovoľného zamestnanca na webovom serveri aplikácie. Ten môže každý deň pridávať rôzne novinky, či zmeny, ktoré sa dejú v ZOO. Pri tvorbe tohto webového servera je využitá ASP.NET aplikácia, ktorá využíva SQL databázu. Po skúšaní na lokálnej databáze bola po menších úpravách nahraná celá webová aplikácia na server a zamestnanec tak získal možnosť rýchlej úpravy. Administrátor, ktorý je zodpovedný za aktuálnosť aplikácie potrebuje informačný systém, ktorý mu ponúkne jednoduchú možnosť aktualizácie.

Pri implementácii však nastali chyby. Po vyskúšaní voľnej domény aj s databázou u rôznych klientov som nedokázal implementovať správne webový server. Funkčný informačný systém je funkčný iba na localhoste.

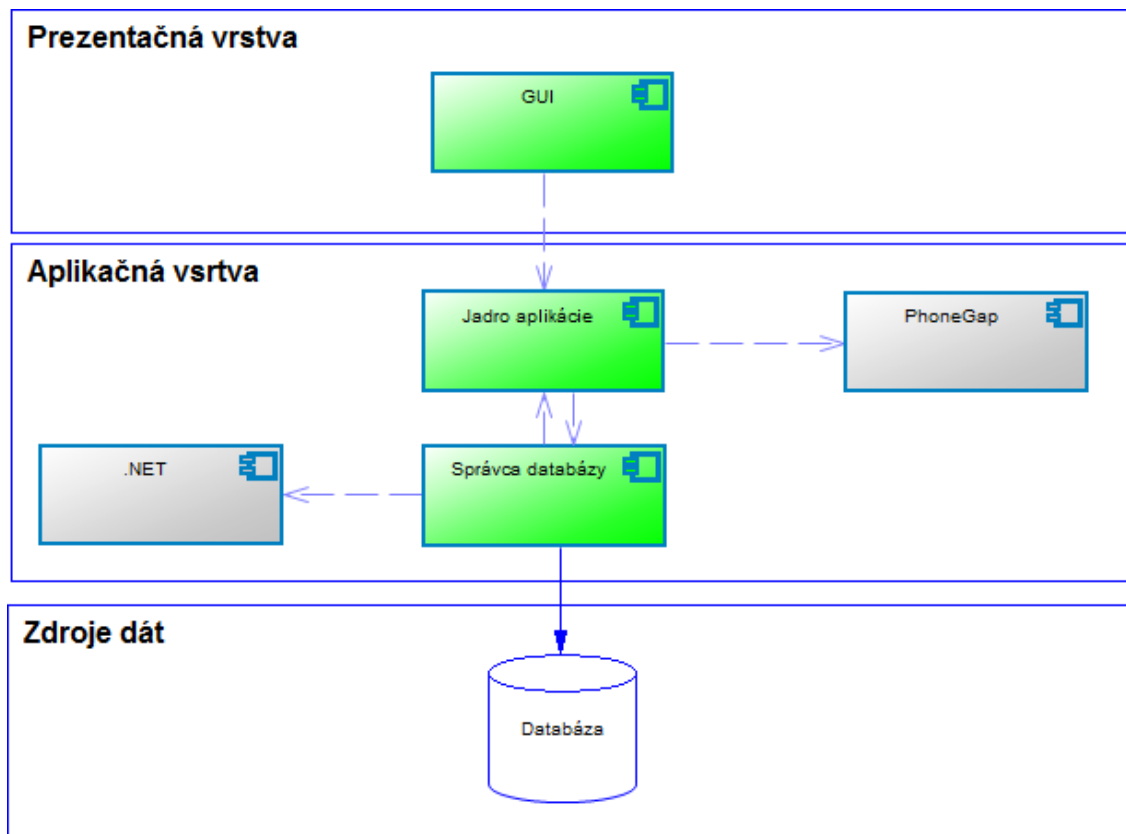
Po prepojení aplikácie s webovým serverom je vývoj aplikácie ukončený. Ich komunikácia je však podmienená pripojením na internet a preto niektoré časti aplikácie nemusia byť aktuálne. V dnešnej dobe to však nerobí veľký problém, pretože užívatelia sú väčšinou pripojení na mobilný internet. Druhou možnosťou je ponúknuť pripojenie na bezdrôtové lokálne siete a ponúknuť tak užívateľom automatické aktualizovanie dát v aplikácii.

5.3.4 Prehľad konečnej aplikácie

V konečnom dôsledku je výsledkom mojej bakalárskej práce aplikácia, ktorá je aktualizovaná zo serveru a dodržiava klasickú trojvrstvovú architektúru. Vzory informačných systémov sú vlastne pravidelne sa opakujúce spôsoby, ako sa programy vyvíjajú. Po logických krokoch, ktoré boli zjavné aj pre bežného človeka sme sa dostali na vysoko špecializovanú časť informatiky, kde vývojári využívajú vzory, ktoré fungujú.

Zistili sme tak, že využívanie vzorov má zmysel a pri rozumnom prepojení s prídavnými modulmi tak vieme získať vo veľmi rýchlom čase kvalitne vypracovanú aplikáciu. Na nasledujúcom obrázku, vid' Obrázek 1.7 Komponenty je vyobrazený diagram komponentov. Pozostáva z troch vrstiev: prezentačnej, aplikačnej a zdrojov dát.

Na prezentačnej vrstve sa nachádza komponent s názvom GUI, čo je grafické používateľské rozhranie. Tento komponent zabezpečuje korektné vykreslenie grafických prvkov aplikácie. Aplikačná vrstva obsahuje správcu databázy, ktorý využíva platformu .NET a pripája sa za pomoci jej metód na databázu. Ďalej sa na tejto vrstve nachádza jadro aplikácie, ktoré využíva softvérový rámec PhoneGap. Potrebné informácie sú potom z jadra aplikácie zobrazené v grafickom rozhraní aplikácie.



Obrázek 1.7: *Komponenty*

5.4 Kompilácia

Po vývoji aplikácie, kvalitnom testovaní a odladení všetkých problémov prichádza na rad publikovanie aplikácie. To zahŕňa zopár krokov, ktoré sú nevyhnutné. Napríklad pri Android aplikácií musíme nastaviť `versionName` a `versionCode` priamo do XML súboru v zložke `"www/config.xml"`, viď Kód 1.2 Konfiguračný súbor aplikácie. PhoneGap aplikácia prepíše celú zložku platforiem, kde použité dáta z webovej aplikácie.

```
<manifest android:hardwareAccelerated="true"
android:versionCode="3" android:versionName="1.0.1"
android:windowSoftInputMode="adjustPan"
package="cz.vsb.bakalaris"
xmlns:android="http://schemas.android.com/apk/res/android">
```

Kód 1.2: Konfiguračný súbor aplikácie

Ďalším krokom je vytvorenie zámku aplikácie. Ten pozostáva z hesla a musí byť bezpečne uložený u vývojára. Pri stratení toho zámku je stará aplikácia nemenná a pri vydaní

novej verzie nás bude Google Store považovať za nového vývojára s novou aplikáciou. Pri zámkach sa však odporúča používať rôzny zámok pre každú vyvíjanú aplikáciu. V prípade rozšírenia vývojárskeho tímu, či odovzdání aplikácie inému vývojárovi je nutné odovzdať zámok. Pri používaní jedného spoločného zámku pre všetky aplikácie by sme stratili bezpečnosť.

Následný bod, prihlásenie vo zverejnenom móde aplikácie. Zložitá zmena nastáva hlavne pri používaní API kľúčov od centrál vývojárov, Google Developers. Pre príklad Google API Android Maps V2 využíva rozdielne SHA1 certifikáty pri aplikácii vo vývoji a pri aplikácii, ktorá je zverejnená. Pri úspešnom vytvorení kľúča, zadaní správnych certifikátov, či API kľúčov môžeme pristúpiť k vydaniu zverejnenia našej aplikácie. Jednoduchšia možnosť pri Android aplikáciách je pomocou Eclipsu. Druhou možnosťou je zadať príkazy do príkazového riadka, vid' Kód 1.3 Postup pri podpisovaní aplikácie. Po úspešnej verifikácii je naša aplikácia pripravená na zverejnenie v Google Store.

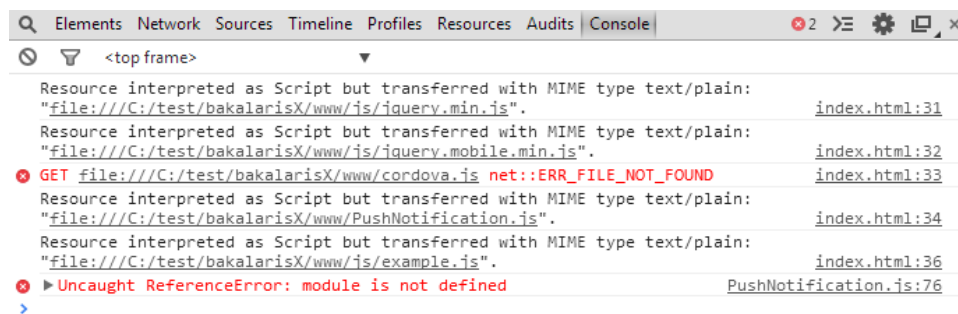
```
ant release  
cd bin  
jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore  
/Users/username/Documents/path/to/my-release-  
key.keystore AppName-release-unsigned.apk app_name  
jarsigner -verify -verbose -certs AppName-release-unsigned.apk  
zipalign -v 4 AppName-release-unsigned.apk AppName.apk
```

Kód 1.3: Postup pri podpisovaní aplikácie

6 Testovanie

Testovanie multiplatformových aplikácií ponúka viaceré riešenia. Prvým je testovanie pomocou emulátora, ktorý ponúka každý mobilný softvérový rámec. Pri porovnaní hlavných emulátorov vidíme na prvý pohľad značné rozdiely. Kým veľmi kvalitné prevedenie od Windows Phone Emulator má nízku náročnosť a nespomaľuje prácu, emulátor od Androidu má často nedostačujúcu rýchlosť načítania. iOS emulátor prináša komplikácie, ktoré sú spoločnosť Apple vlastné. Nutnosť zaplata licencie pre iOS emulátor, ako aj nepodporovaný najrozšírenejší operačný systém v počítačoch, Windows.

Druhá možnosť nám ponúka využitie webového prehliadača na počítači. Táto možnosť však ignoruje natívne riešenia a tak nám tento druh testovania veľa nepovie. Pre vývojára je zaujímavé testovanie užívateľského rozhrania a taktiež je možné testovanie JavaScript kódu, vid' Obrázek 1.8 Chyby vo Web Inspectore. Ten je veľmi náročné odladiť v spustenej aplikácii na mobilnom zariadení a aj to jedine vďaka ďalším externým nástrojom. Vďaka službám ako Web Inspector od Google Chrome, alebo Firebug môžeme odladiť JavaScript kód, vidieť XHR žiadosti, či prezrieť DOM elementy.



Obrázek 1.8: Chyby vo Web Inspectore

Tretiu možnosť testovania vidím v kompilácii aplikácie na mobilné zariadenie. Veľmi rýchla odozva, podpora všetkých natívnych prvkoch, ako aj autentická odozva jednotlivých prvkov robia z tejto možnosti jednu z najrelevantnejších. Problémom je široká škála zariadení a mobilných operačných systémov. Testovanie tak je obmedzené na zariadenia, ktoré sú dostupné.

6.1 Testovanie pri vývoji

Od začiatku vývoja bola mobilná aplikácia ZOO Ostrava testovaná a odchyťované všetky jej nedostatky. Dôležitá bola maximálna funkčnosť a spoľahlivosť každého prevedeného kroku vývojára. Problémom testovania pri vývoji je fakt, že prebehla na troch zariadeniach Androidu, či emulátore a preto už pri vývoji mohli nastať nepresnosti.

S pribúdajúcim časom a rastom aplikácie boli pri testovaní zachytené rozdielne chovanie aplikácie na rôznych zariadeniach. Multiplatformové aplikácie majú najväčšiu slabinu rýchlosť prevedenia operácie a v tomto smere sa zatiaľ natívnym aplikáciám nemôžu rovnať. Už pri vývoji som pochopil, že finálna aplikácia bude musieť byť maximálne jednoduchá, aby spĺňala požiadavky užívateľov. Vynútila si to mierna zmena pôvodného návrhu či obmedzenie dynamickosti aplikácie, aby boli aj menej výkonné zariadenia schopné využívať služby. Minimálnu hranicu pre aplikáciu bola nastavená na verziu 2.2 - Froyo.

6.2 Užívateľské testovanie

Testovanie finálnej verzie aplikácie môže predísť problémom a potvrdiť funkčnosť aplikácie pre koncového zákazníka. Pre vývojárov, ktorí často pracujú na veľkých informačných systémoch je bežné automatické testovanie. Pri multiplatformových aplikáciách je táto možnosť daná vďaka externým nástrojom, ako je Selenium. Problém však nastáva pri zmene užívateľského rozhrania, kde doterajšie výsledky strácajú na význame.

Druhou možnosťou je manuálne testovanie. Keďže aplikácie v mobilných telefónoch nezvyknú mať veľkú kapacitu a taktiež problémy s rýchlosťou hybridných aplikácií, manuálne testovanie sa mi javí ako ideálne riešenie.

Pri manuálnom testovaní sú za potreby ľudí, ktorí testujú a zariadenia, na ktorých sú testy prevádzané. Je nevyhnutné, aby testovanie malo pevný základ a vychádzalo z rozobratia jednotlivých situácií. Zatiaľ čo vývojár je ovplyvnený časom, ktorý strávil pri aplikácii, všima si zložitejších vymožeností aplikácie a jednoducho prehliadne vážne problémy. Preto je asi ideálnym riešením aplikáciu na testovanie ponúknuť pravým užívateľom, teda tým, ktorí budú aplikáciu používať.

Riešenie problémov zariadení je u iOS pomerne jednoduché. Zopár verzií zariadení od Apple nerobí veľký problém na testovanie skoro všetkých zariadení, od iPhone až po iPady. Väčší problém je napríklad v Android zariadeniach, ktoré majú rozdielny výkon, rozdielne rozlíšenie, hustoty displeja a často aj mnohé softwarové vylepšenie. Preto sa vývojár snaží testovať Android zariadenia, ktoré reprezentujú všetky dôležité skupiny Androidu.

6.3 Zhodnotenie vlastností aplikácie na rôznych platformách

Veľmi jednoduchá aplikácia je tým najlepším, čo môžeme do mobilného telefónu ponúknuť. Človek, ktorý sa nachádza na mobilnom zariadení má vo väčšine prípadov záujem informovať sa a pokračovať vo svojej práci.

Aplikácia napriek svojej jednoduchosti pri využívaní rôznych prídavných modulov zachytila rozdiely v rýchlosti pri špičkových mobilných zariadeniach a pri bežných múdрых telefónoch. Nevyhnutným bodom pri vytvorení multiplatformovej aplikácie je tak neustála optimalizácia. Táto časť v oblasti vývoja aplikácií sa vyvíja veľkou rýchlosťou a každé zrýchlenie behu aplikácie je dôležité.

Dôležité optimalizácie multiplatformových aplikácií:

- Regenerujte užívateľské rozhranie na server
- Obmedzte internetový prístup
- Nečakajte na dáta pred zobrazením užívateľského rozhrania
- Používajte CSS prechody a hardwarovú akceleráciu
- Vyhnite sa zdržaniu pri kliknutí
- Optimalizujte obrázky
- Obmedzte tieň a prechody
- Minimalizujte spätné toky
- Využívajte softvérové rámce, ktoré fakt potrebujete
- Testujte

[6]

Pri dodržiavaní týchto základných bodov a sledovaní moderných trendov v oblasti multiplatformových aplikácií môžeme dosahovať skvelé výsledky a dokonca sa vyrovnávať natívnym aplikáciám.

Záver

Bakalársku prácu ohľadne multiplatformových aplikácií hodnotím ako skvelú voľbu. Začiatočná nevedomosť vo vývoji týchto aplikácií sa zmenila. Je jednoznačné, že jeden projekt nemôže posunúť mladého programátora do pozície profesionála. Vidím veľké rezervy v mojich vývojárskych zručnostiach, no vďaka tejto práci som urobil veľký skok. Práca na projekte od začiatku, prvotnej myšlienky, či nákreсу, až po finálnu podobu, ktorá je inšpiráciou širokej škále zariadení, je veľmi prospešná pre ďalšiu kariéru. Aplikácia je funkčná na webových prehliadačoch, mobilných zariadeniach či tabletoch všetkých druhov, systémov, či značiek.

Hlavný prínos vidím v teoretickej pripravenosti vo svete mobilných aplikácií. V súčasnosti mi nebude robiť problém sledovať aktuálne novinky a využívať ich správnym spôsobom. Dostať aplikáciu do stavu, že ostáva odoslať registračný poplatok a som schopný nahrať svoju aplikáciu na Google Store, je príjemné. Osobný pocit je aj zaväzujúci, že moja práca, či teoretické alebo praktické vzdelávanie v oblasti mobilných aplikácií, by mala pokračovať.

Projekt Ostravská ZOO napriek svojim nedostatkom zvláda konkurovať najlepším aplikáciám v oblasti zoologických záhrad. Aplikácie zo San Diega, Columbusu, či čiastočnej sekcie z pražskej ZOO neponúkli žiadnu inšpiráciu, ktorú by som do Ostravskej ZOO musel implementovať. Aplikácia však potrebuje optimalizáciu. Hlavným nedostatkom je súbežne zvyšovať kvalitu či precíznosť užívateľského rozhrania a taktiež znižovať náročnosť aplikácie, teda zrýchlenie jej chodu.

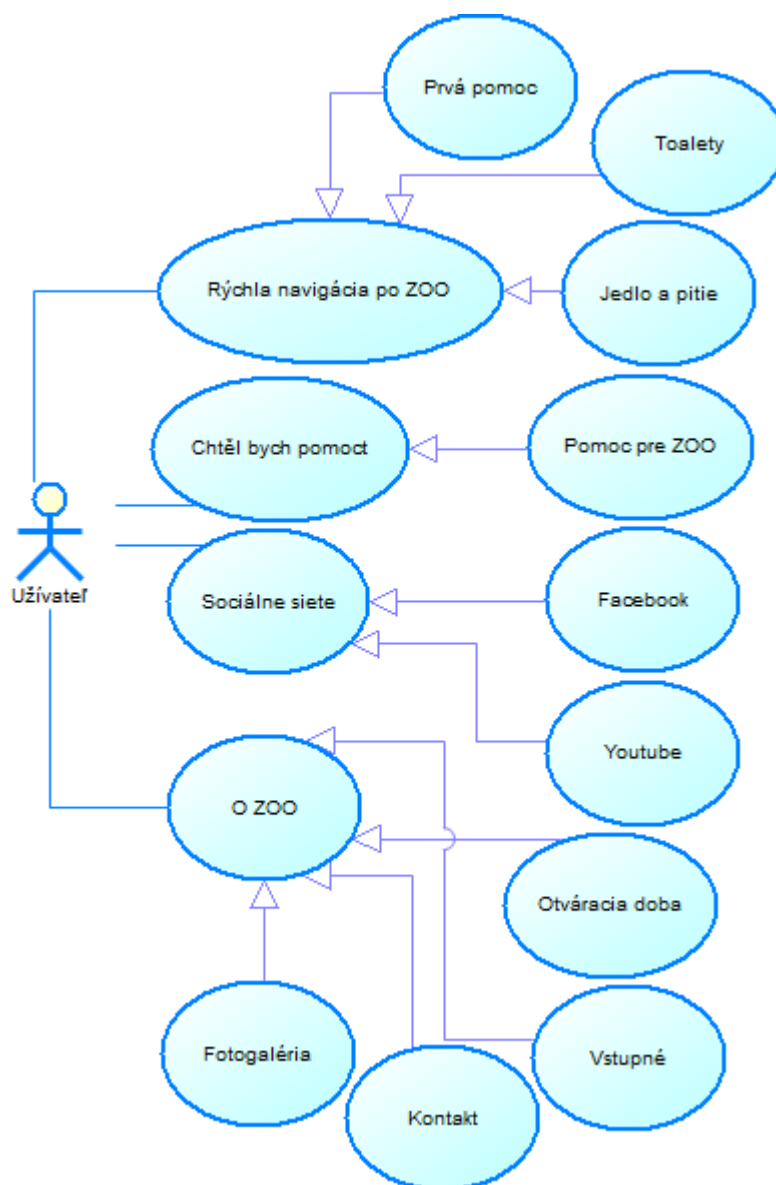
Multiplatformové aplikácie sú skvelou voľbou pri vývoji mobilných aplikácií a ich najväčšie výhody, ktoré sú cena, čas, či využívanie známych vývojárskych prostredí priláka veľké množstvo vývojárov.

Použitá literatúra

- [1] CHOVAŇÁK, MICHAL. História verzií Androidu [Infografika]. In: Môj Android: Najčítanejší portál o Androide na Slovensku [online]. <http://www.mojandroid.sk/historia-android-verzii-infografika/>. 2013 [cit. 2014-07-16]. Dostupné z: <http://www.mojandroid.sk/historia-android-verzii-infografika/>
- [2] Get Started with Publishing. Google Developers [online]. 2012 [cit. 2014-07-16]. Dostupné z: <http://developer.android.com/distribute/googleplay/start.html>
- [3] IOS Developer Program. APPLE. Apple Developer [online]. 2014 [cit. 2014-07-16]. Dostupné z: <https://developer.apple.com/programs/ios/>
- [4] Account types, locations, and fees. Windows Dev Center [online]. 2014 [cit. 2014-07-17]. Dostupné z: <http://msdn.microsoft.com/en-us/library/windows/apps/jj863494.aspx>
- [5] WARGO, John M. PhoneGap essentials: building cross-platform mobile apps. Upper Saddle River, NJ: Addison-Wesley, c2012, xxiv, 359 p. ISBN 03-218-1429-0.
- [6] Christophe Coenraets: Web Platform, Cloud and Mobile Application Development. Top 10 Performance Techniques for PhoneGap Applications [online]. 2013 [cit. 2014-07-16]. Dostupné z: <http://coenraets.org/keypoint/PhoneGap-performance>

Zoznam príloh

Príloha A: Využitie informácií v prílohe



Súčasťou dodávky distribuovanej na CD/DVD médiu sú:

- zdrojové kódy aplikácie
- spustiteľná verzia aplikácie
- dokumentácia aplikácie
- samotná bakalárska práca
- readme.txt pre rýchly návod na spustenie aplikácie